

**REMARKS/ARGUMENTS**

The present amendment is submitted in accordance with the Revised Amendment Format.

The Examiner has rejected claim 1-3, 5, 6, 9-13, 15-20, 23 and 24 under 35 Application under 35 U.S.C. § 103(a) as being unpatentable over Sheshadri, "Understanding JavaServer Pages Model 2 architecture," published: December 1999, pages 1-140, ("Sheshdari") in view of U.S. Patent No. 6,654,932 B1 to Bahrs ("Bahrs"), and further in view of Prosise, "ASP.NET: Web Forms Let You Drag and Drop Your Way to Powerful Apps," pages 1-28 ("Prosise").

The Examiner has rejected claim 4 of this Application under 35 U.S.C. § 103(a) as being unpatentable over Sheshadri, in view of Bahrs and Prosise, and in further view of Leeche, 4GuysFrom Rolla.com, published: December 1, 1999, pages 1-5.

The Examiner has rejected claim 7 of this Application under 35 U.S.C. § 103(a) as being unpatentable over Sheshadri, in view of Bahrs and Prosise, and in further view of U.S. Patent No. 6,981,215 B1 to Lindhorst et al. (hereinafter "Lindhorst").

The Examiner has rejected claims 8, 14, 17, and 21-22 of this Application under 35 U.S.C. § 103(a) as being unpatentable over Sheshadri, in view of Bahrs, Prosise, U.S. Patent No. 6,453,356 B1 to Sheard et al. (hereinafter "Sheard"), Goodwill, "Pure Java Server Pages," published: June 08, 2000, pages 1-4, and in further view of U.S. Patent No. 6,331,187 B1 to Jeyaraman.

No new matter is added.

**Rejection under 35 U.S.C. § 103(a)**

The Examiner rejected claims 1-24 as being unpatentable under 35 U.S.C. § 103(a) in view of various combinations of cited references. For the following reasons,

Applicants contend that the references relied on by the Examiner, as understood, do not render these claims obvious.

**Claims 1, 13, 16**

The central issue in this case is whether independent claims 1, 13, and 16 are unpatentable under 35 U.S.C. § 103(a) over Sheshadri, in view of Bahrs, and in further view of Prosise. Applicants submit that these claims are not unpatentable at least because a combination of Sheshadri, Bahrs, or Prosise still does not disclose each and every element recited in the claims. Claims 1, 13, and 16 recite a method including an “object” with specific functions and properties. The use of such “objects” is not contemplated by any of these references. For example, claim 1 recites:

“A computer program product, tangibly embodied in a machine-readable storage device, the computer program product comprising instructions operable to cause data processing apparatus to perform operations comprising:

- providing a server-side framework to an application, the server-side framework being external to the application, the framework supporting predefined data types, each data type having a predefined rule;
- receiving from an application a request for an object, the request indicating one of the multiple predefined data types, the object storing a default value of the indicated data type, the default value being stored in the object in a transfer format and in a process format, the process format being different from the transfer format;
- creating the object in response to the request;
- generating a markup language page that includes the default value in the transfer format read from the object;
- sending the markup language page to a browser on a client;
- receiving a user-supplied value in the transfer format from the browser;
- replacing in the object the default value in the transfer format with the user-supplied value in the transfer format, the object automatically converting the user-supplied value from the transfer format to the process format, the object storing the user-supplied value in the process format, the object automatically checking the compliance of the user-supplied value in the process format with the predefined rule; and
- if the user-supplied value in the process format complies with the predefined rule, forwarding the user-supplied value in the process format from the object to the application and otherwise automatically resending the markup language page to the browser with the user-supplied value in the transfer format.”

(Claim 1)(Emphasis Added).

Claim 1 includes nine (9) references to a particular object that operates in the context of a server-side framework of a client-server system between an application and a browser on a client. Based on the express language of claim 1, the object includes each of the following:

- The object is created in response to a request from an application;
- the object stores a default value having a data type indicated in the request;
- The object stores the default value in both a transfer format and a process format;
- The object is used to generate a markup language page (the default value in the transfer format read from the object into the page)
- The default value in the transfer format in the object is replaced with the user-supplied value in the transfer format from a browser on a client;
- The object automatically converts the user-supplied value from the transfer format to the process format;
- The object stores the user-supplied value in the process format; and
- The object automatically checks the compliance of the user-supplied value in the process format with a predefined rule

Sheshdari does not disclose such an “object.” The Examiner mistakenly relies on “CD objects” disclosed in Sheshdari as disclosing the “object” in claim 1. However, the CD objects in Sheshdari are substantially different than the “object” of claim 1. CD objects are Javabeans that are created in response to requests from the client, not the *application* as in claim 1. Sheshdari, p. 7, “Everytime a user adds an item within EShop.jsp, the request is posted to the controller servlet. The servlet... instantiates a new CD bean.” This is a telling distinction. Because the request in Sheshdari originates from the user, it does not and cannot indicate a data

type, where the “object” is used to store a default value having the data type indicated in the request from the application as claimed. Moreover, the listing for the CD bean relied on by the Examiner is shown in Listing 4. Unlike the “object” of claim 1, the CD bean in Sheshdari does not store data in both transfer and process formats, does not convert a user-supplied value from the transfer format to the process format, and does not automatically check the compliance of the user-supplied value in the process format with a predefined rule. The Examiner’s reliance on Sheshdari is entirely misplaced.

Further, Bahrs does not disclose such an “object.” While Bahrs does disclose a “validation object”, such a validation object is substantially different than the “object” of claim 1, and further is in an entirely different context. For instance, Bahrs is in the field of “creating client applications.” Bahrs, col. 1, lines 10-15. More specifically, Bahrs deals with creating a “client development architecture.” Bahrs, col. 2, lines 41-42. The Background section of Bahrs describes the need for an improved client development environment (e.g., for thick and thin clients). In contrast, the language of claim 1 specifies that the “object” is a server-side construct (e.g., the request that is used to create the object is from an application on the server-side). However, the Examiner has strung together citations in Bahrs that are unrelated to the disclosure of claim 1, are further unrelated to each other, and are unrelated to server-side operations. For example, using Bahrs, the Examiner ties together col. 5, lines 8-30 in the Summary with col. 16, lines 1-20 in the Detailed Description. However, col. 15, lines 35-37, which puts the Examiner’s citation in context, specifically states that “[t]he architectural pattern of the present invention is illustrated as a Java implementation for building thin (or thick) client applications...” Validation rules on a client are substantially different than checking compliance of data from a client on a server using an “object” as set forth in claim 1. Accordingly, the Examiner’s reliance on Bahrs is also entirely misplaced.

Finally, Prosise is entirely deficient. Prosise does disclose that “retain[ing] their state across postbacks,” but as understood there is absolutely no disclosure as to how this occurs. Prosise is a magazine article showing high level features of ASP.net, but does not disclose an “object” or the details as set forth in claim 1. Prosise is a completely deficient reference.

Given all the deficiencies described above, one of ordinary skill in the art would not have considered claim 1 obvious under 35 U.S.C. § 103(a) over a combination Sheshadri, in view of Bahrs, and in further view of Prosise. These references do not teach or suggest all the limitations of claim 1. Further, these references are not properly combined because at least Bahrs addresses a different technical problem.

For the above reasons, Applicants contend that claim 1 is not unpatentable under 35 U.S.C. § 103(a). Claims 13 and 16 include substantially all the same limitations as claim 1. Therefore, these claims are patentable as well. The dependent claims in this Application are patentable at least because they include all the limitations of these three independent claims and additional limitations.

### CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 408-244-6319.

Respectfully submitted,

/Chad R. Walsh, Reg#43235/

Chad R. Walsh  
Reg. No. 43,235

FOUNTAINHEAD LAW GROUP P.C.  
900 Lafayette Street, Suite 200  
Santa Clara, CA 95050  
CRW  
Tel: 408-244-6319